

PROFESSOR ALLOCATION SYSTEM

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

(Bhargavi Dogra (151401))

Under the supervision of

(Dr. S.P Ghrrera)

To



Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Wagnaghat, Solan-173234, Himachal Pradesh

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled “ **PROFESSOR ALLOCATION SYSTEM**” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Vknaghat is an authentic record of my own work carried out over a period from August 2018 to May 2019 under the supervision of **(Dr. S.P Ghrera)** (Head of the Department -Computer Science and Engineering).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)

Bhargavi Dogra, 151401.

This is to certify that the above statement made by the candidates is true to the best of my knowledge.

(Supervisor Signature)

Supervisor Name :Dr. S.P. Ghrera

Designation : Head of the Department

Department name: Computer Science and Engineering

Dated:

Acknowledgement

I here, feel very grateful presenting this project report on “**Professor Allocation System**”. I am very fortunate here to have support of number of key individuals during this project report. Hereby taking chance to thank them all for helping me all the way during the study report.

First of all, I would like to thank my project guide **Dr. S.P Gharera**, Head of the Department of Computer Science and Engineering ,for helping me and giving tips on how to proceed with this project and report. A heart full of gratitude for devoting his valuable time and illuminating suggestions and ideas on how this project could be enhanced and improved in many occasions during this project study .This thoroughly helped us in attaining in depth understanding of concepts and varied technologies used in my project.

I would also like to extent a very profound and warm thanks to all those people whose name may not have been appeared in here but whose contribution has not gone unnoticed. I am personally very thankful to all of them.

PROJECT REPORT INDEX

Candidate's Declaration	i
Acknowledgement	ii
ABSTRACT	1
1. Chapter-1 INTRODUCTION	2 - 6
1.1 Introduction	
1.2 Problem Statement	
1.3 Objectives	
1.4 Methodology	
1.5 Organization	
2. Chapter-2 LITERATURE SURVEY	7 - 11
Literature Survey	
3. Chapter-3 SYSTEM DEVELOPMENT	12 - 31
Model Development	
• Analytical	
• Computational	
• Experimental	
• Mathematical	
• Statistical	
4. Chapter-4 Algorithms	32 – 40
5. Chapter-5 Test Plan	41 - 50
4. Chapter-6 RESULTS AND PERFORMANCE ANALYSIS	51 - 60
• Analysis of system developed either	
• Results	
• Output	
• Analysis of above results	
5. Chapter-7 CONCLUSIONS	61
5.1 Conclusions	
5.2 Future Scope	

PROFESSOR

ALLOCATION

SÝSTEM

BHARGAVÍ DOGRA (151401)

ABSTRACT

The primary accentuation of this project is to build up a productive web empowered application that can computerize the requirements of representative data. The executives which incorporate data of leave subtleties of each educator and assignment of lecture sets to other professors, off that chance in case if somebody is inaccessible at a specific point of time. This is done for computerizing the current manual framework for fast data preparing and to fill in as an archive of information for better data trade and future recovery.

Modular view controller is picked to structure such a product. It confines the application rationale from the utilized interface layer and supports detachment of concerns.

Here the controller gets all solicitations for the application and afterwards work with model to set up any information needed by the view. The view at that point utilizes the information arranged by the controller get ready to create the last respectable reaction.

The algorithm proposed sets up the scheduling of lecture to other faculty members in fair means in case if someone is absent.

Chapter-1 INTRODUCTION

1.1 Introduction

The plan and usage of a Professor Allocation System shows a UI so as to supplant the paper records which are right now being utilized. School Staff will almost certainly get to all perspectives through online entry gave on the web.

Primary emphasis of this project is to develop an efficient web enabled application that can automate the needs of employee information management which include information of leave details of every professor or employee and allocation of lectures if someone is unavailable at a particular point of time.

The system is used to provide user authentication, to displaying only mandatory information as per the individual's requirement. Whole data is completely reviewed and verified on the server before any kind of alteration occur which can only be done by the administrator. Along with staff user interface, this system is capable of planning admin user interface, allowing admin to use it to add, alter and delete faculty data into the data base. Complete information is put away safely on the servers being overseen by the school head and ensures most noteworthy conceivable dimension of security. This framework contains a logging framework to follow every one of clients' information and offers adjustment to get to information rules and is guaranteed to expand the productivity of the school's record the executives along these lines diminishing the heap expected to get to and convey address to understudies.

Thus, our project is aimed at automating the existing manual system for quick information processing and to serve as a repository of data for better information exchange and future retrieval. The goal is for developing and implementing the system cost effectively. The data should be well protected for personal use and should be capable of making the data processing fast. The system's objective of easily maintaining the information can then be achieved.

1.2 Problem Statement

Whenever a faculty proceeds on leave, relief is arranged so that classes are not affected Professor Allocation System will read the time table file available on 172.16.73.6 and find the available faculty who can take the class. It is also to be ensured that load on account of frequent absence of a faculty does not fall on the same faculty every time.

Professor Allocation System will also inform all concerned about the class engagement schedule. In addition, the system will perform all standard Professor Allocation System functions like addition of faculty data by the admin, showing of faculty daily lecture schedule along with the extra lectures allotted incase of another faculty is on leave. The lecture allocation for subjects whose faculty is on leave is done on the basis of round robin scheduling algorithm where faculty who is having the prior experience of the subject along with the qualifications and empty lecture slots shall be allocated to take up the lecture of the faculty on leave.

1.3 Objective

Project's objective is to:

1. Explore and report the approaches to Leave Management Strategies.
2. Implement and demonstrate a considered approach for the above that works for the JUIT system.
3. Perform comparative analysis for performance of various approaches.

The development consists of the following activities, which help us to automate the whole process keeping in mind the database integration approach.

1. The application is expected to be user friendly.
2. The application is expected to be flexible.
3. Application is expected to provide us with the extreme level of security with distinct level of authentication.

There should be no risk of data ineptly at any level while the project is under development phase and post launch.

1.4 Implementation Methodology

Model View Controller is opted to design this software. It separates the logic of our application from the UI layer.

Here, the controller receive the coming requests and works with model to make the data needed by view. View thereafter makes use of the data consolidated by the controller in order to formulate the final presentable response.

The Model-View-Controller design divides the complete responsive application into three main components model, view and controller.

The **model** comprise of the main data and functionality of our application. **View** displays the information which the user instills on front end and **Controllers** manage the input given by the user. Controllers and view collaborate in order to provide the UI.

Model and Controller and View provide for an interactive application in three parts namely user input, processing part, and output part. The Model wraps back-end part and functionality and is independent of input behavior and relative output functionality.

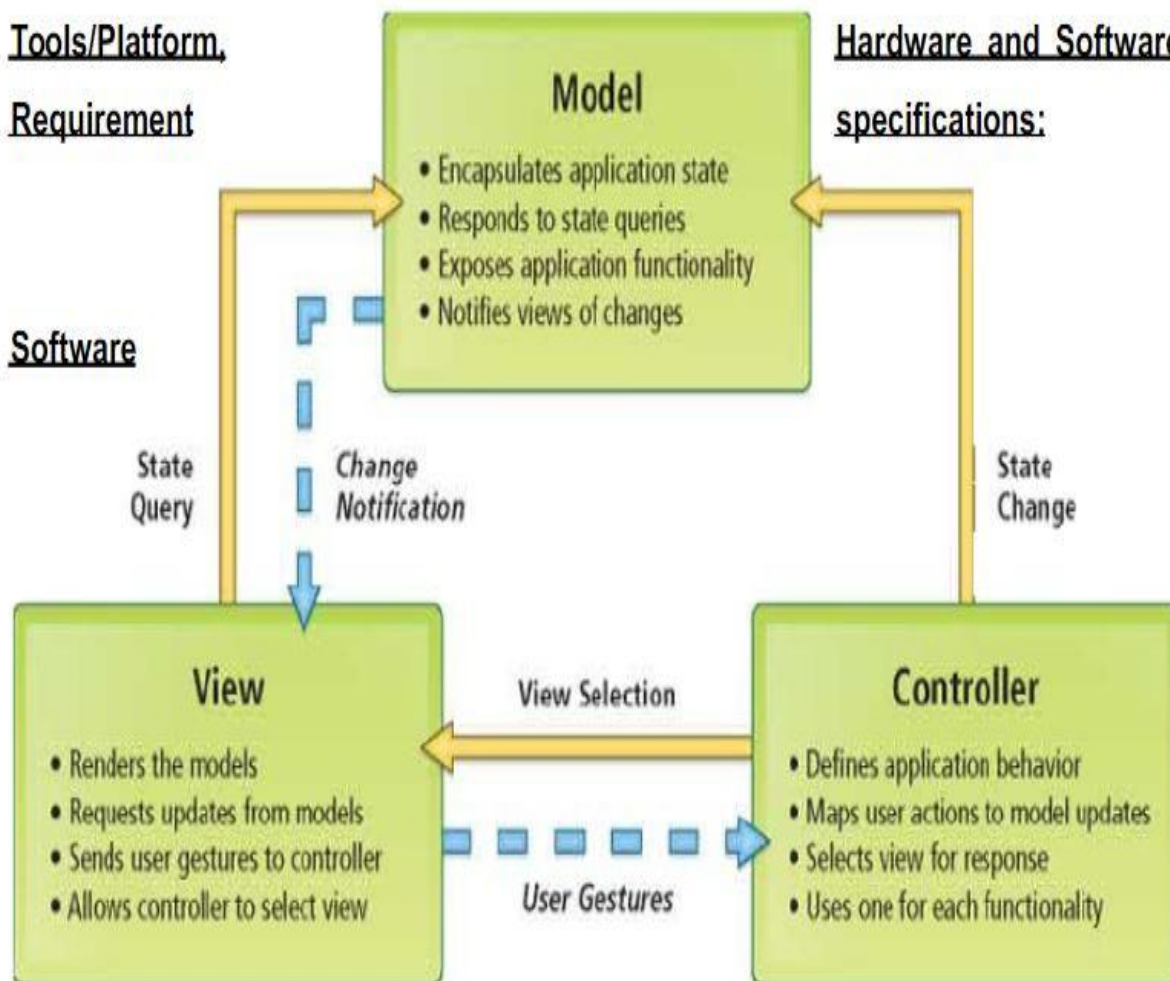
View components display the contents to the customer. A view gets contents from modal. There may be one or more views of a single model. Each of them is connected to the controller component. Controller shall get input from the person using such application, usually events that accept input from keyboard and mouse movements. Requests are converted to eventss for the view or the modal. Controller shall only be responsible for providing the user interaction and interface.

DATA FLOW DIAGRAMS

Tools/Platform,
Requirement

Hardware and Software
specifications:

Software



Method Invocations Requirements:

Events

Chapter-2 LITERATURE SURVEY

Writing sneak peak is an approach to assess and comprehend the inquires about prior in a specific zone. The basically center is to yield a combined and calculative revaluation of the theme through different justifiable techniques'. The principle objective here is to locate a pertinent report with respect to booking calculations to give a legitimate and smooth lecture allotment to another educator if there should be an occurrence of an employee being on leave.

“A Visual Implementation of Student Project Allocation”

Authors – Mhd. I. Massa and A.E-Atta

“(International Journal of Computer Theory and Engineering, April- 2011)”

<https://pdfs.semanticscholar.org/0dc8/a616dc1824c1059686e880e9bdc633fa957f.pdf>

In this paper understudy venture assignment is being given inclination over sets which was an utilization of the stable coordinating issue.

The understudies supply inclination records over ventures that were offered by instructors and every teacher supplies an inclination list over understudies who show enthusiasm for at least one of his tasks. Understudy Project Allocation (SPA) is a two-sided coordinating issue on the grounds that the contribution of SPA is a two disjoint sets inferring the arrangement of understudies and set of ventures, and we try to coordinate individuals from set of understudies to individuals from set of tasks subject to different criteria. This paper exhibits another information structure for the understudy venture assignment issue with inclination records over (understudy, venture) sets to decrease the space that expected to display an occasion of SPA-(s, p). In SPA-(s, p) the understudies supply inclination records over activities, and the instructors supply inclination records over (understudy, venture) sets. This accomplishes a harmony between understudies which lessen the quantity of unmatched understudies by keeping understudy from being segregated.

SPA-(s, p) defeats these weaknesses in light of the fact that the instructors can be twinned among understudies and tasks, the speaker 1 may favor student1 to work in certain activities, in a similar time he lean towards different understudies to work in different undertakings.

In SPA model instructor give his inclination over understudies, so on the off chance that he lean towards an understudy to another, at that point he will favor the equivalent in all ventures he advertised. In the event that the speaker supplies inclination over sets, the understudy gets an opportunity to work in one of the ventures offered by teacher subject to similar criteria. Then again; SPA-p model gives inclination over tasks with lack of concern between the understudies, which may deny the understudies to work with their favored undertakings. Be that as it may, SPA-(s, p) works lack of concern and it works too towards the desires of understudies and it stays away from startling un-assignments.

“Analysis of Adaptive Round Robin Algorithm and Proposed Round Robin Remaining Time Algorithm”

Authors- Arjita Sharma, G. Kakhani

“(International Journal of Computer Science and Mobile Computing, Vol.4 Issue.12, December-2015, pg. 139-147)(Online available at www.ijcsmc.com)”

The Round Robin Scheduling Algorithm is presented essentially for time sharing framework. Each procedure is appointed a period interim which is fixed with the end goal that procedure needs to execute in that time interim, called time quantum, for which it is permitted to run. The Round Robin Scheduling calculation is a reasonable booking calculation that gives equivalent time quantum to all procedures. The decision of the time quantum is significant as it modifies the calculation's execution. The investigation of this paper incorporates Adaptive Round Robin

Algorithm and Proposes another calculation Round Robin Remaining Time Algorithm which centers around improving execution of Adaptive Round Robin Algorithm as far as Average Waiting Time (AWT) and Average Turnaround Time (ATT).

The new proposed calculation that orchestrates the procedures in climbing request of burst time, and after that it picks the savvy time cut (STS), which is for the most part dependant on the quantity of procedures. It is a Priority Driven Scheduling calculation dependent on burst time of procedures. At first, it masterminds the procedures as indicated by the execution time/burst time in expanding request that is the littler the blasted time, the higher the need of the running procedure. The subsequent stage is to pick the brilliant time cut (STS), which is mostly dependant on the quantity of procedures.

brilliant time cut = burst time of the mid procedure, (number of procedures : odd)

brilliant time cut (STS)= normal of the CPU burst of every single running procedure (number of procedures: even)

Analyses and estimations results are utilized to fundamentally settles the calculation in fixed time quantum (which is viewed as a test for Round Robin Scheduling Algorithm). Calculation's fundamental suspicions incorporate that all procedures land in the meantime in the prepared line. In 2013 ,Manish Kumar Mishra concentrated on improving the Round Robin Scheduling Algorithm instituted upgrading CPU execution utilizing the highlights of Shortest Job First and Round Robin booking with fluctuating time quantum. The calculation proposed is tentatively turned out to be superior to Standard Round Robin Algorithm. The reproduction results demonstrate that the holding up time and turnaround time have been decreased in the proposed calculation contrasted with conventional Round Robin.

Following is the proposed Round Robin Remaining Time Algorithm:

Stage 1: Assign Process to prepared line.

Stage 2: Rearrange all the procedure in expanding request of their burst time

Stage 3: While (prepared line! =NULL)

Stage 4: Calculate time quantum = $\sum p_i / 2 * n$

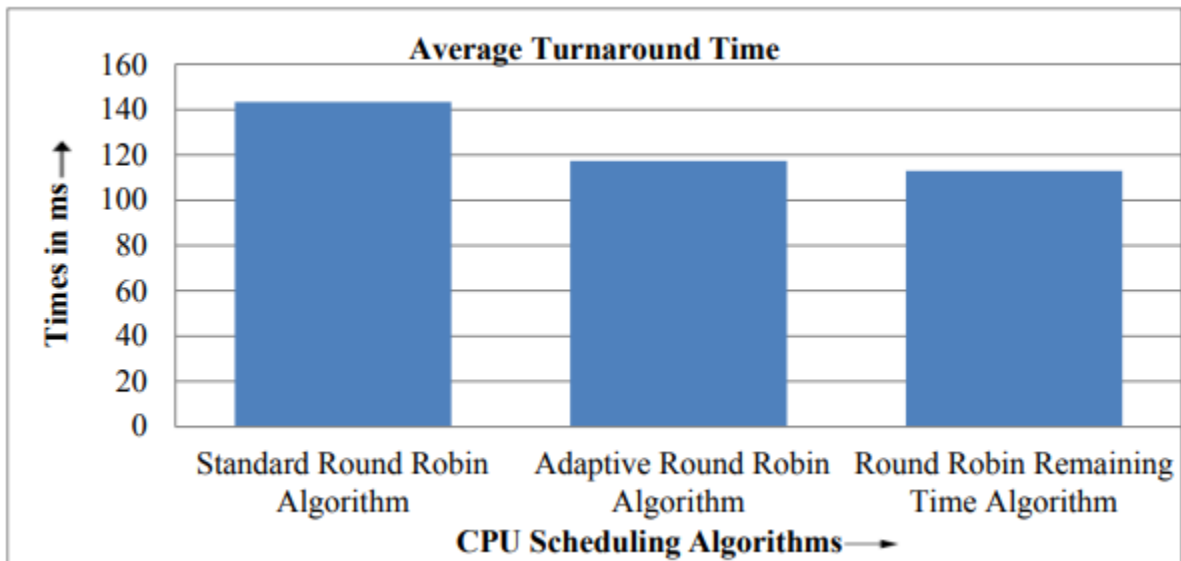
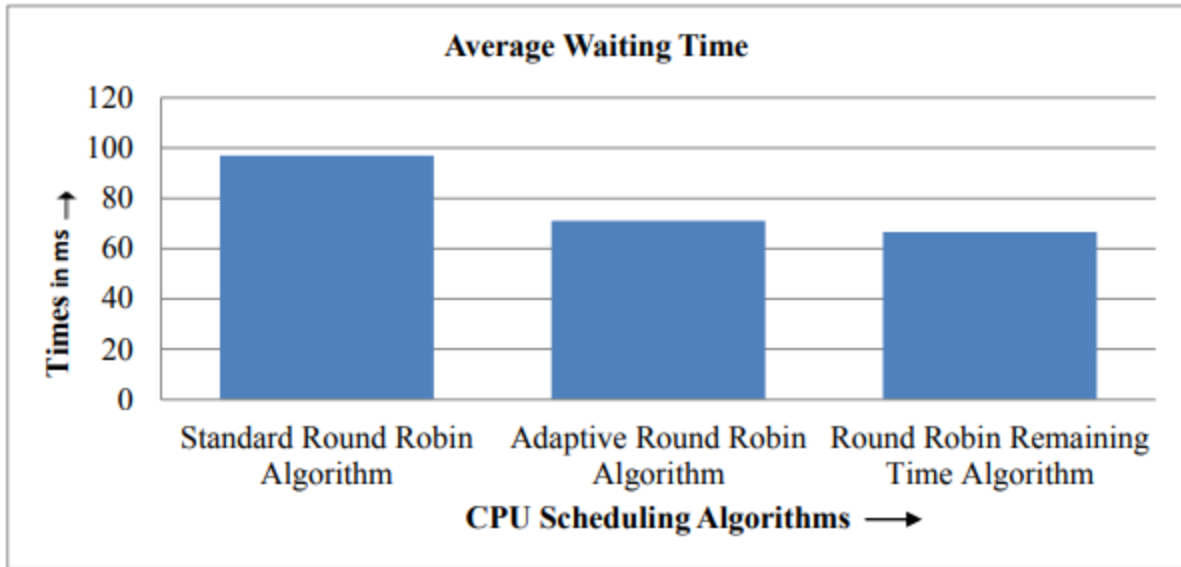
Stage 5: If (remaining burst time < time quantum) Allocate CPU again to the present running procedure for outstanding burst time Else Remove the present running procedure from the prepared line and put it at the tail of the prepared line.

Stage 6: If no of procedure > 0 Go to stage 5

Stage 7: End while

Stage 8: Calculate normal holding up time, normal turnaround time

RESULTS



Chapter-3 SYSTEM DEVELOPMENT

The System Development Methodology meant is considered to be as a structure implemented for planning and controlling the method of creating the special information system. Highlighting the process of software development is the major concern, rather than proper planning for the purpose of developing the highly integrated and specific software. The persistency of the procedures is to provide a compact software development as per the requirement.

Right Software development choice for life cycle development

Choice of Software Development Life Cycle methodology is a hard task for developers and software engineers. The difficulty here is that only a few organizations know the actual criteria to select a methodology in order to add value to the software. In order to consider a framework to select a given SDLC methodology, it is required to define its different types and demonstrate its pros and cons of those models.

How to select the correct SDLC

Selecting the appropriate SDLC is a process that one can implement internally or consult for. To choose the right model, we have gone through the following steps:

- Learn about SDLC Models
- Access the needs of stakeholders
- Define criteria

STEP 1: Learn about the SDLC Models

SDLCs are the same in their usage. In order to select the right SDLC, we should have enough experience and be familiar with the SDLCs that will be chosen and understand them correctly. Models are similar to the tools that important to know each tool usage to know which context it can fit into.

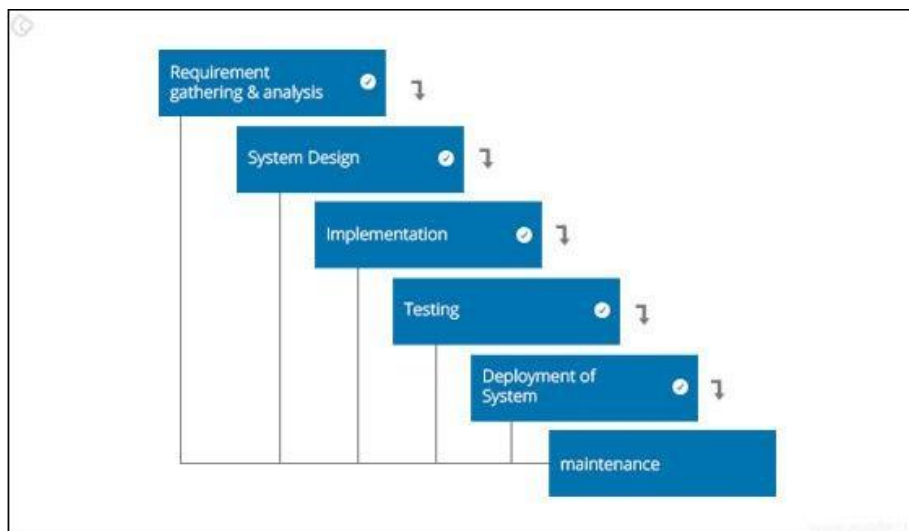
To get through various models, we have analyzed following SDLC models and analyzed advantages and disadvantages:

Waterfall Model:

On the off chance that we are into programming improvement eventually or the other, we would have caught the Waterfall Model.

Considered as the customary technique for clarifying the product improvement process in programming building, cascade model happens to illuminate the procedure into a straight stream with a predetermined succession to give the clients a chance to comprehend that further dimension is made dynamic on finish of the past one.

Besides, this system likewise discusses the way that returning to manage the progressions is preposterous.



Advantages:

1. Easy to comprehend and work all Sufficiently
2. simple to deal with as model is inflexible
3. Saves huge measure of time
4. Allows for simple testing and investigation

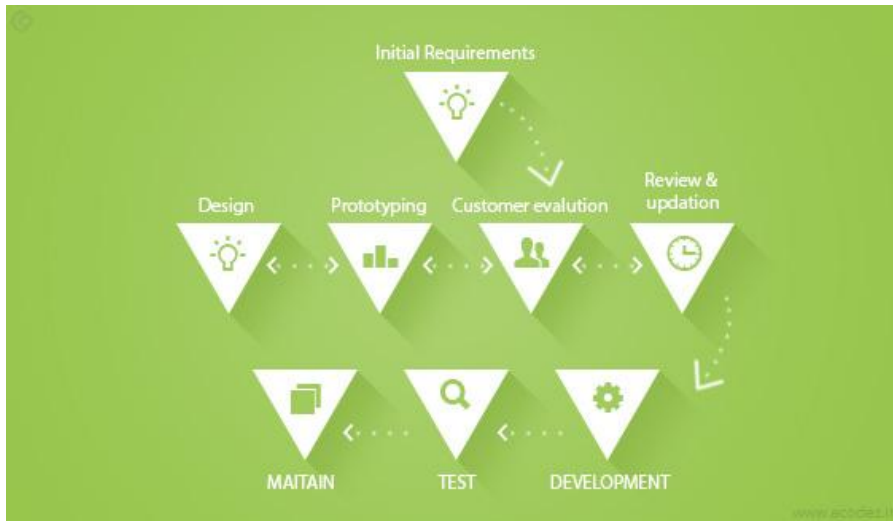
Drawbacks:

1. Only matches exact requirements
2. Not appropriate for support ventures
3. No alternative to know conceivable result of a venture
4. Not great for long and continuous tasks

Model Methodology:

It is a particular programming improvement strategy that starts designers towards making just the example of the goals to approve its utilitarian pith to the clients and roll out fundamental improvements before making the bona fide last arrangement.

Truth be told, the best piece of this technique is that it will in general purpose a lot of differentiating issues happening with the cascade strategy.



Focal points:

1. Gives clear thought regarding the practical procedure of the product
2. Reduces the danger of disappointment in a product of uncertainty
3. Assists well in necessity gathering and the general investigation

Detriments:

1. Chances of increase in the executives cost
2. Extreme association of customer can influence preparing
3. Workflow of the product gets influenced by such a large number of changes.

The Software Development Methodology(Agile):

The spray programming advancement methodology, innovatively is utilized for articulating efficient task association strategy taking into consideration rehashed adjustments.

Without a doubt, such sort of a philosophy is one speculative layout for undertaking different programming building ventures.

Any increasingly beneficial thing about it is that it limits hazard by making programming in brief time, known as emphases , which end up lasting from multi week to one month.

Advantages:

1. Adaptive approach that responds to changes is constructively
2. Allows for straight communication to be maintained lucidity
3. The Enhanced quality by discovery and fixing the defects swiftly and identifying probability mismatches before the time.

Disadvantages:

1. Focus on working with the software and lacks documentation competence mainly.
2. Chances of being getting off target as outcomes are ambiguous

Quick Application Development:

Implied at giving speedy results, the previously mentioned application development is planned to give extraordinary advancement forms with the assistance of other improvement approaches.

It is delivered to exploit from the advancement programming.

Unquestionably, it is intended to grow the usefulness of the entire programming improvement methodology for featuring the support of an active user.



Advantages:

1. Makes the complete development for the process natural
2. Assist it for client in taking smooth previews
3. Encourage response from the customers for enhancement

Disadvantages:

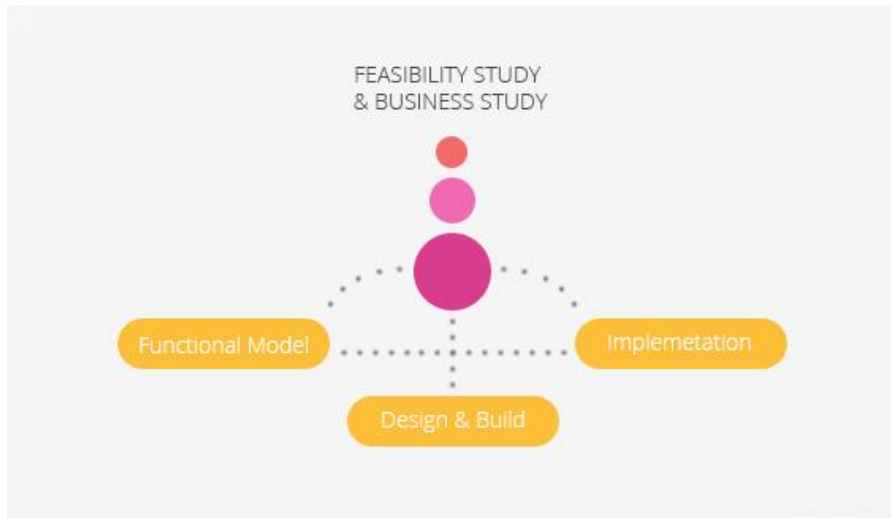
1. Dependent on the team for presentation
2. Works on modularized system confine on this methodology
3. Requires really skilled workforce to handle complexities
4. Not applicable for the tiny budgeted projects

Dynamic System Development Model Methodology:

Legitimately defined and got from the quick application improvement system, it is an iterative and steady methodology that centers around the commitment of the client.

The undertaking of this technique is to give programming improvement frameworks inside the exact time period and the assigned spending plan.

The motivation behind why it is beautiful sought after in the realm of programming advancement.



Advantages:

1. Users be getting a clench of the software, the development process
2. Functionality for the deliverables are immediate
3. Offers easily right of entry to end user by the developers

Disadvantages:

1. The methodology is pricey to put into practice
2. Not apt for undersized organizations

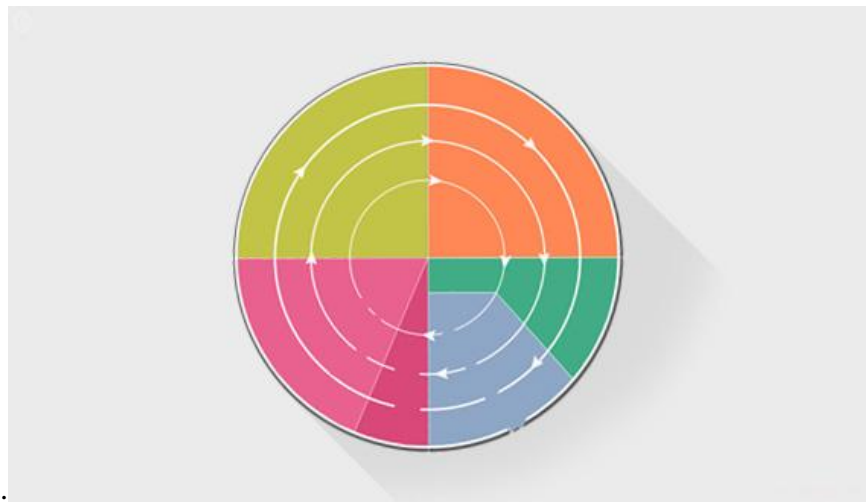
Spiral Model:

Being an awfully complex design, it is meant to moderate the early threats in the development.

As per the process going by, the developers initiate onto a minor level and explore the included risks in it.

Further too this, the developers are envisioned towards crafting a plan for iterating of the spiral.

The achievement of any Spiral Lifecycle model is based on reliable, vigilant, and conversant



administration of the project.

Pros:

1. Risk factors are considerably abridged
2. Excellent for the large and complicated projects also
3. Allows for supplementary functionality too
4. Suitable for highly on risky projects with the varied business need also

Cons:

1. Exorbitant model in programming improvement
2. Failure in hazard investigation stage may harm the entire task

3. Not suitable for generally safe endeavors might to get proceeded and never additionally wrap up Intense Software design Approach:

Serious writing computer programs is perceived by the way that customer commitment in the product improvement process is incredibly high.

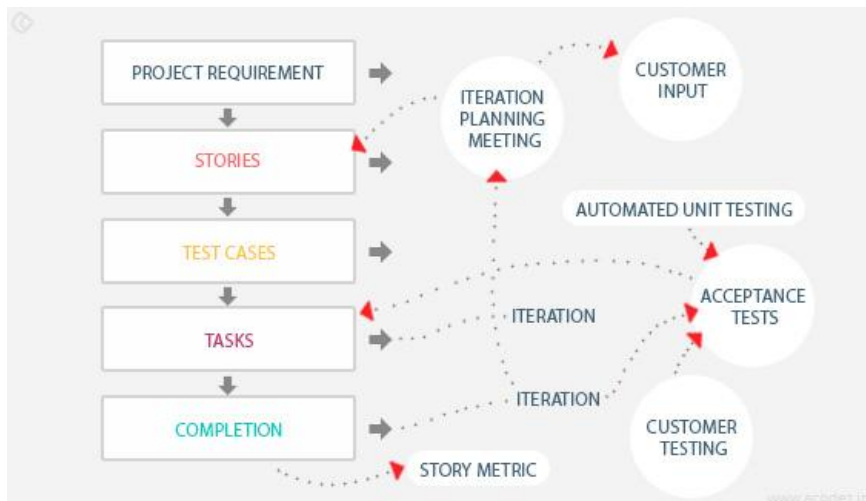
As a light-footed programming building approach, extraordinary programming strategy is by and by known as technique.

It is fundamentally used for including programming inside an unequal situation.

It enables progressively imperative tractability inside the appearing.

The basic purpose of this model is to lessen the cost of programming fundamentally.

It is truly partaken in the model that the expense of changing the necessities on future stage in the assignment can be really hollering.



Pros:

It sets center around shopper support

1. Establishes normal plans and timetables
2. Developers are exceptionally committed to the venture
3. Equipped with vanguard approaches for quality programming

Cons:

Productivity relies upon the general population tangled

1. Requires repetitive gathering for advancement lauding all out expense
2. Necessitates for exorbitant improvement
3. Exact chances and up and coming results are truly not known

We will take through a portion of the extra qualities that we can appreciate when utilizing:

Correspondence

In XP, the procedure of correspondence is straightforward, vigorous and genuinely straightforward. Every one of the colleagues is reliant on each other and offer data inside the group – which implies that every part knows about the job and usefulness of the other.

Effortlessness:

Since the association organize itself starts with a straightforward and clear methodology, effortlessness is ensured over every other stage. Also, in this unique circumstance, effortlessness alludes to executing a methodology where we cut all the poo and incorporate just the vital data.

Criticism

With criticism, it is simpler to find the territories where improvement can be achieved alongside adjustment in the systems that are being connected here.

Inspiration

Daringness is only a lot of activities whenever utilized that may be harmful to the group and the business prerequisites that will be performed. Along these lines, with inspiration, we would now be able to deal with detaching those indispensable elements that may influence our administrations.

Extraordinary programming interfaces five particular people in a group and these incorporate client, coordinator, programmer, quality checker and a tracker..

Stars:

1. It lays center around client cooperation
2. Establishes intelligible plans and timetables
3. Developers are amazingly steady quick to the venture
4. Equipped with hello tech techniques for predominance of the product

Cons:

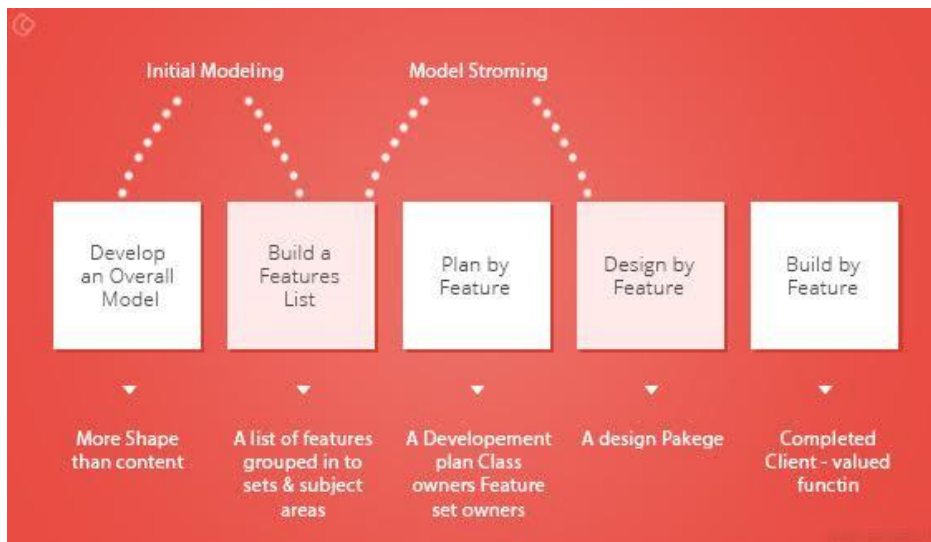
1. Worth relies upon the general population included
2. Requires intermittent gathering for improvement raising all out expenses
4. Strict potential outcomes and future results are extremely unknown

Highlight Driven Development:

Being an iterative methodology for programming improvement, it is away for serving a significant number of gatherings wearing down an endeavor subject to article arranged development.

This sort of model is decent for associations that are passing on from a phase based strategy to an iterative system.

It is starting at now known as a method and is exceedingly utilitarian and adequately innovative to oversee changed complexities.



Geniuses:

1. Move to bigger undertakings with uninterfered with progress
 2. Simplest 5 methods gets result a superior smooth way
 3. Built on pre-set criteria of programming advancement, it is customized for simple improvement
 4. Projects that need consistent updates are worked by highlight driven improvement that guarantees all needs are taken care of.
 5. Results in highlights that dependably eclipse the data sources
5. Since this depends on the absolute best programming advancement in practices, any engineer with the significant experience would be able to deal with and deal with the venture related errands effortlessly.

Cons:

1. Not reasonable for little tasks and a solitary designer – dependably a tremendous group is required, which implies that we can't ever ensure an energetic due date.
2. Highly trustworthy on the main engineers, requiring for the total structure – the procedure should be observed through each stage as even a moment blemish can make disarray in the framework.

3. Highly trustworthy on the main engineers, requiring for the total structure – the procedure should be checked through each stage as even a moment blemish can make disarray in the framework.

Joint Application Development Methodology:

The Joint Application Development Methodology is a prerequisites grouping and UI extension approach that in requires for the end-clients, customers and engineers go to an amazing off-site meeting to emphasize and affirm programming framework.

This approach serves towards incorporating the customer in the structure and extension of an application.

This is immediately practiced through a progression of intentional workshops known as sessions.

It tends to lay unquestionable quality on the business exertion instead of prepared nuances.

Permits with respect to concurrent in assembly and coalition of additional data.

1. Produces the huge volume of significant data in brief period
2. Proximate settling of abberations with appropriate aid
3. Provides discussion to investigate different focuses

Cons:

Takes exorbitant measure of time for arranging and

Requires earth shattering venture of time and Calls for positively prepared specialists, which is strong top find

The Lean Development Methodology:

As a specialized progression, Lean Development model accentuation on the arrangement of easily reasonable programming.

This unpredictably planned advancement strategy is more intentionally immersed than some other type of light-footed technique.

The target of this methodology is to improve the product in 33% of the time, with confined spending plan, and less measure of basic work process

Aces:

1. Lower spending plan and lesser time prerequisites
2. Allows for conveyance of item early

Cons:

1. The usefulness of the group chooses achievement of software advancement of process

Rational Unified Process Methodology:

Keenly called as RUP, Rational Unified Process philosophy powers programming improvement utilizing judicious instruments.

This strategy isolates the development procedure into four unique stages that each incorporates business displaying, examination and plan, authorization, testing, and demeanor.

This is an item based and web-engaged program development procedure.

The model will in general help programming designer for expressing rules, formats, and examples for all highlights and phases of programming improvement.

Pros:

1. Lays high spotlight on exact documentation
2. Removes task dangers connected with customer developing needs
3. Extremely insignificant need for combination

Cons:

1. Needs exorbitantly master programming designer
2. Development strategy of the technique is muddled
3. Amalgamation may cause perplexity
4. Very tangled to get it

Scrum Development Methodology:

SCRUM is the extensively favored dexterous programming improvement strategy.

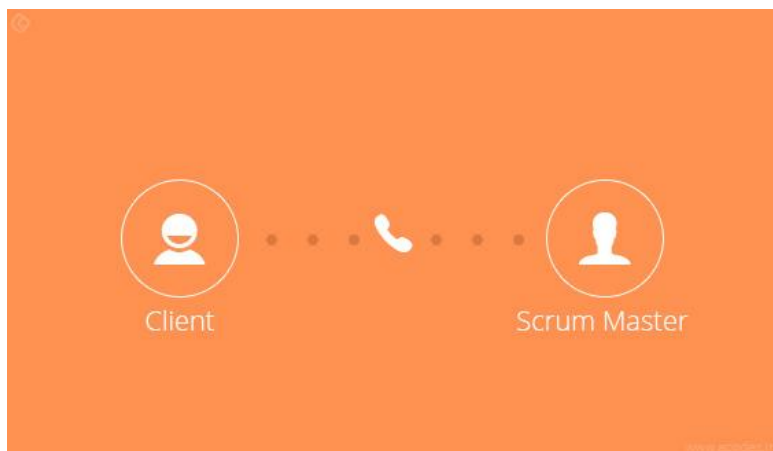
(So also, KANBAN is a training that encourages groups to participate and work effectively.)

On a very basic level, this brilliant improvement is appropriate for those advancement extends that are continually shifting or immensely creating necessities.

The Scrum Software improvement model starts with a transient arranging, gathering and finishes with an end survey.

This development practice is utilized for brief improvement of programming that happens to incorporate a progression of emphases to produce required programming.

It is an ideal methodology since it easily expedites track the intentional advancing ventures.



Pross:

Evaluation lies in the hands of the group

Business prerequisite document(BRD) is Considered to be - >trivial

1. Lightly controlled technique relating to consistent refreshing

Conss:

1) The preparing strategy laments because of the faltering expenses

2) Inappropriate for enormous measured ventures

STEP 2: Assess the needs of Stakeholders

Following Criteria are analyzed:

- User Requirement: The initial requirement is clear but the overall requirements are unclear.
 - o *Suitable Methodologies: Agile; Spiral*
- Familiarity with Technology: We are familiar with Spring, REST Services, and other technologies to be used
 - o *Suitable Methodologies: Waterfall; Agile*
- System Complexity: System to be designed is expected to be simple and abstracted.
 - o *Suitable Methodologies: Agile, Waterfall (not best suited, but can be used)*
- System Reliability: System to be designed is expected to be fault-tolerant and reliable
 - o *Suitable Methodologies: Waterfall, Agile (not best suited, but can be used)*
- Schedule: Short-time schedule
 - o *Suitable Methodologies: Agile; Iterative*
- Cost Limitation: limited
 - o *Suitable Methodologies: Agile; Iterative*
- Visibility of Stakeholders: Visibility high
 - o *Suitable Methodologies: Spiral; Agile*
- Component Reusability: Components should be reusable
 - o *Suitable Methodologies: Waterfall; Iterative*

Considering above analysis, we considered the trade-offs of various methodologies which we could live with and some pros that could help us fast-track the development process and decided to go with *Agile Software Development Methodology*.

STEP 3: Define the Criteria

We have defined few selection criteria or arguments to verify if our selection is right. Following are the questions which give weight to our decision:

- **Is the Software Development Life Cycle appropriate for the size of our team and their skills?**
 - *Yes, Agile suits a smaller team.*
- **Is the Software Development Life Cycle appropriate for the selected technology we use for implementing the solution?**
 - *Yes, excellence in different technologies would complement each other while development through Agile methodology.*
- **Is the Software Development Life Cycle appropriate for client and stakeholders concerns and priorities?**
 - *Yes, direct communication in Agile suits it.*
- **Is the Software Development Life Cycle appropriate for the geographical situation (distributed-team)?**
 - *Not Applicable*
- **Is the Software Development Life Cycle appropriate for the size and complexity of our software?**
 - *Yes.*
- **Is the Software Development Life Cycle appropriate for the type of projects we do?**
 - *Yes*
- **Is the Software Development Life Cycle appropriate for our software engineering capability?**
 - *Yes*

- **Is the Software Development Life Cycle appropriate for the project risk and quality insurance?**
 - *Yes, testing at each phase ensures this.*

Chapter-4 ALGORITHMS

Booking is named as an idea in PC performing various tasks and multiprocessing working framework plan, and progressively working framework structure. It fundamentally alludes to the manner in which forms are allocated needs in a need line.

There are few scheduling areas which we can use to compare scheduling algorithms:

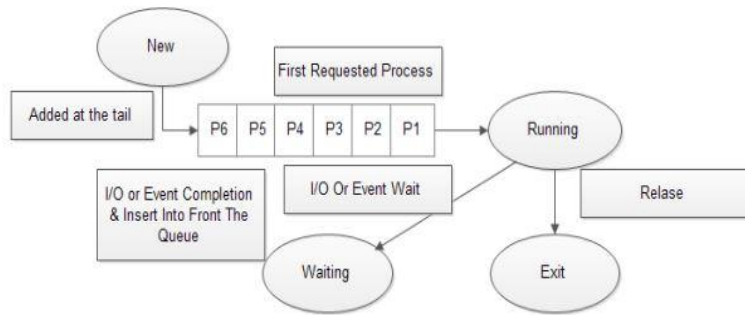
“CPU utilization, Throughput, Waiting time, Response time, Fairness, Starvation, Predictability, Preemption”

Study of various Scheduling Algorithms

FIRST COME FIRST SERVE (FCFS)

First-Come-First-Serve algo is called as the most straightforward planning calculation. The procedure is arranged here based on their entry time on the prepared line. Due to its arrangement as non-preemptive planning algo, it races to culmination when it's admitted to the CPU. That is the reason, this is alluded to as keep running until done and utilizes FIFO calculation to finish the doled out errand. In interchange words, it infer that one program will run non-preemptively until it is done. In this way, the FCFS construction isn't significant in planning intuitive clients' becoz of the way that it can not ensure great reaction time.

Easy to engrave and straightforward; its significant downside is that the avg. time of this calculation is typically long. Likewise, the FCFS isn't reasonable in the manner that bigger employments make littler occupations pause and non significant bigger employments make significant littler employments pause. Throughput is extremely less and this is a direct result of the way that the bigger procedure need a bigger time to be executed, which makes the CPU work without any assistance.



Most brief JOB FIRST (SJF)

The scheduler composes forms with exceptionally less evaluated preparing time staying in the line. The above booking takes a shot at the procedure with the most brief burst time.

It is the appropriate way to deal with limit holding up time.

Types:

- Non Pre-emptive
- Pre-emptive

To effectively execute it, the term time of the procedures ought to be intended to the processor ahead of time, which is essentially unrealistic all over the place.

The above scheduling algorithm is most appropriate if every one of the occupations are mostly available in the meantime.

Problem with Non Pre-emptive SJF:

If where section time for systems are not same, which infers all of the methods are not open in the readied line at same time, and a couple of occupations get in contact after some time, ordinarily technique with humbler burst time must be obliged to anticipate the present methodology's beginning to complete, in light of the fact that in non pre-emptive SJF, on arriving of a strategy with tinier period, the present work's running isn't stopped to execute the little business starting. This circumstance results in the matter of Starvation, wherever a little system needs to anticipate a reached out till the present longer method gets dead. This circumstance turns out to be conceivably the most significant factor happens if shorter businesses keep coming.

Pre-emptive SJF:

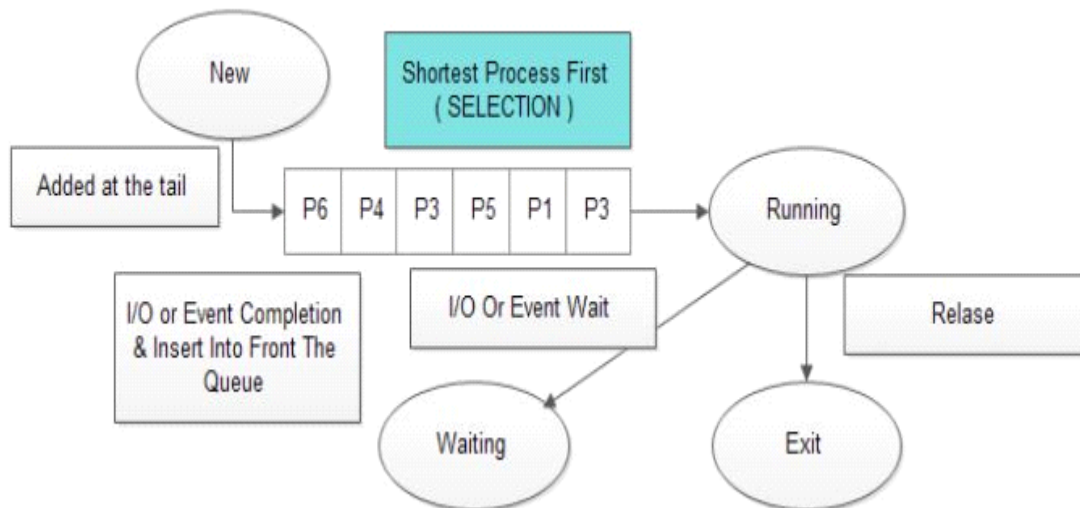
Here employments are made to go into prepared line when they arrive, yet as an Aprocess with the littler burst Time arrives, the present procedure Sis fired for quite a while from being executed, and the shorter activity is permitted to execute before.

Advantages:

SJF is a reasonable calculation for the procedure that has a less execution time. Regarding FCFS, it's holding up time and avg. time is progressively worthwhile. This happens in light of the fact that the execution of little procedures in begin brings a less time sitting tight for long procedures.

Disadvantages:

One of the fundamental downside of the referenced calculation is: a procedure must hold up in the line on the off chance that it has a bigger execution time despite the fact that it might be in the queue for quite a while. Intricacy is the no. of bugs of the referenced calculation which remains in the determination of the following procedure from the CPU.



PRIORITY BASED SCHEDULING

It is a non-preemptive calculation and a standout amongst the most normally utilized planning calculations. Each procedure is appointed a need. The procedure with greater need is to be executed first, etc further. Procedures with same need are executed on first started things out serve premise. Need can be chosen dependent on memory prerequisites, time necessities or some other asset necessity.

Need planning can be either - preemptive or non-preemptive.

At the point when a procedure lands good to go line, its need is contrasted and the need of right now running process. A non-preemptive need planning calculation will basically begin another procedure at head of prepared line.

Hindrances:

The serious issue with need booking is starvation. In need planning some low need procedure may continue hanging tight uncertainly for CPU. In a vigorously stacked system, continuous entry of higher need procedures can avoid low need process from getting the CPU. In this condition two things will occur, it is possible that we can run the procedure when framework turns out to be daintily stacked, or the PC framework will crash and lose all incomplete low need forms.

One answer for the issue of starvation is maturing.

Maturing is a strategy of continuously expanding the need of procedure that sits tight in the framework for a more extended timeframe.

Eg:- If need go from 125(low) – 0(high) we could decremented the need of holding up procedure by like clockwork.

In the long run when a procedure with an underlying need of 127 will have a most noteworthy need in the framework and it would be executed.

ROUND ROBIN SCHEDULING (RR)

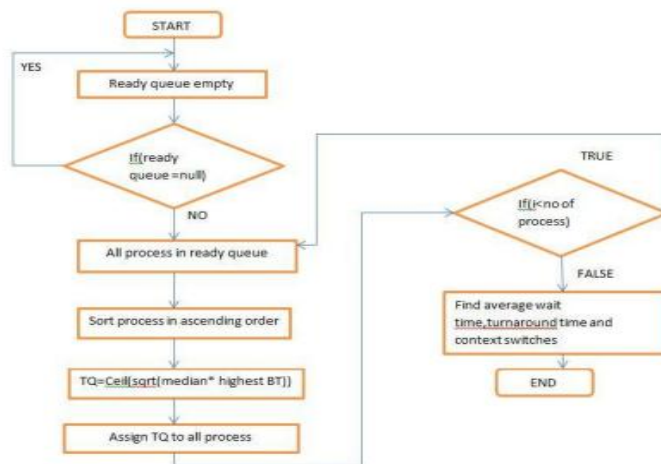
In spite of the fact that FCFS plot is possibly terrible for short employments; Round Robin conspire tackles this issue by giving each procedure a little unit of CPU time which is known as quantum time which fluctuates from 10 – 100 milliseconds. Round Robin is additionally one of the least difficult planning calculations for procedures in a working system. In actuality, when quantum lapses, the procedure is seized and added as far as possible of the prepared line.

The significant preferred standpoint of Round Robin is reasonableness where each activity gets an equivalent measure of the CPU. And its downside is the normal holding up time. The normal holding up time will be hazardous especially once the measure of procedures is gigantic.

The Standard Round Robin Scheduling Algorithm is structured particularly for time sharing framework. Each procedure is allotted a period interim, called its time quantum, amid which it is permitted to run. Each procedure is given a fix time to execute called time quantum. The arranged line is treated as a roundabout queue. The CPU PC equipment goes round the readied line, assigning the CPU to each procedure for a period interim of up to 1 time quantum.

Versatile Round Robin Scheduling Algorithm utilizes most limited burst approach dependent on Smart Time Slice. It is a Priority Driven arranging algorithmic program upheld burst time of processes. First of all we have to organize the procedures as indicated by the execution time/burst time in expanding request i.e. smallest the burst time higher the need of the running strategy.. Round Robin Remaining Time Algorithm forms are allocated to the prepared line based on burst

time and after that rework all the procedure in expanding request of their burst time.



COMPARISON OF FCFS WITH ROUND ROBIN(RR) ALGORITHM

FCFS is not apt for time sharing systems whereas for time sharing systems, Round Robin algorithm is apt.

FCFS is non preemptive whereas RR is preemptive.

There is no priority i.e. processes are executed on first cum first serve basis. RR is not a priority based algorithm.

For the time equivalent to CPU burst time, the process is executed in FCFS whereas in Round Robin, the process is executed for quantum time after which it has been preempted.

FCFS uses a FIFO queue technique while Round Robin uses FIFO circular queue technique.

FCFS has low CPU utilization while RR has medium CPU utilization.

FCFS is unfair for larger jobs which make short jobs wait and non important long jobs make mandatory short jobs wait whereas Round Robin is fair.

In FCFS, there is no potential for starvation whereas in RR, there is free starvation.

Correlation OF SJF WITH ROUNDROBIN ALGORITHM

For time e-sharing frameworks, SJF isn't able though Round Robin is adept for time sharing frameworks.

SJF is both preemptive and non preemptive while RR is just preemptive.

SJF is a need based calculation in which need is backwards of CPU blasted while RR isn't a need based calculation.

Procedure is executed for the time equal to CPU burst time in SJF while in RR process is executed for quantum time after which it is seized.

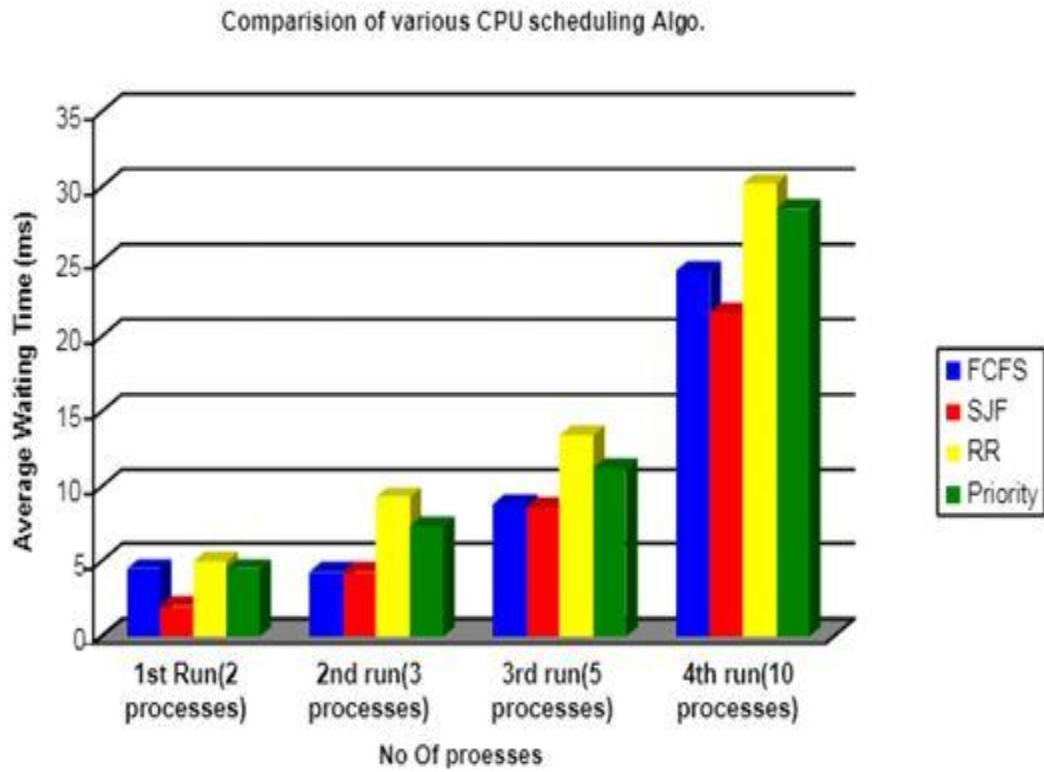
SJF utilizes a need line procedure while utilizes FIFO round line method.

FCFS has higher CPU use while RR has medium CPU usage.

SJF is out of line while RR is reasonable calculation.

It is difficult to anticipate the amt. of CPU time a vocation has left in the event of SJF though it is unsurprising if there should arise an occurrence of RR.

GRAPHICAL ANALYSIS



Chapter-5 TEST PLAN

Intention of Proposed Test Plan

A test plan is a type of agreement of work between Quality Assurer, the software development engineer, and the manager of the product being formed. The goal of the test plan is to provide SDE's input to Quality Assurer and traditional documentation for considering a reference as required. We can make use of previous test plans again as a source for the former documentation, training, or evidence of traceability, as per the need.

Test plan's an incentive in Agile Development

The test plan is the understanding of work which is utilized to portray how the testing is done among improvement and Quality Assurance. It demonstrates the comprehension of the referenced ideas and gives us the documentation of the application. Regardless of whether the referenced plans are constrained to just a couple of pages and comprise just of the obligatory data, nimble groups can even now advantage themselves from such plans.

An effectively decipherable and succinct test plan gives us the preferred standpoint that when the colleague's change or when the group develops then they make progress toward the need of documentation. of use. just as past. discharge data.

Deft gathering capacities as a lone gathering towards a commonplace focus of achieving Quality. Swift Testing has shorter time designations called emphasess. This framework is moreover called release, or transport driven procedure since it gives an unrivaled desire on the practical things in a word range of time.

Nimble Test Strategy

In the Agile Test plan/technique, test plan is composed and refreshed in each discharge.

□ QA is a lot of exercises expected to guarantee that items must fulfill client prerequisites in a precise and in a progressively dependable manner.

□ In SCRUM (coordinated) QA is the obligation of the analyzers as well as of everybody. QA incorporate every one of the exercises we do to guarantee right quality amid the improvement of new items.

Test designs in nimble incorporates:

1. Testing Scope

2. New functionalities which should be tried

3. Level or Types of testing dependent on the multifaceted nature

4. Burden and Performance Testing

5. Foundation Consideration

6. Relief or Risks Plan

7. Resourcing

8. Expectations and Milestones

Project's testing life cycle consist of 4 stages:

1. Start the undertaking or Iteration

Amid first stage, we perform starting setup errands. It incorporates distinguishing individuals for testing, introducing instruments for testing, planning assets (convenience testing lab), and so on. The accompanying advances are pursued to accomplish this progression:

- a) Establishing a business case for the undertaking
- b) Establish the undertaking degree and the limit conditions
- c) Outline the key necessities and use cases that will drive the structure exchange offs
- d) Outline at least one hopeful models
- e) Identifying the hazard
- f) Cost estimation

2. Construction Iteration

The significant piece of the testing happen in this specific phase..To.build an increment.of the arrangement, this stage is observed.as a lot of emphasess. So as to do the equivalent talked about errand, the group need to actualize a blended work on utilizing XPs, Scrums, Agile modelings, and agiles information, etc concerning every emphasis.

In this stage, nimble groups pursue, the organized the necessity practice: With every cycle .they take the most significant prerequisites remaining from the work. thing stack and execute. them.

The referenced stage is separated into 2 – corroborative testing and analytical testing. Verifying testing centers around watching that the structure fulfills the enthusiasm of the accomplices as depicted to the gathering to date, and is performed by the gathering. While the logical testing recognizes the issue that substantiating gathering have skipped or neglected. In Investigative testing, analyzer chooses the potential issues as disfigurement stories. Quick testing oversees ordinary issues like blend testing, load/stress testing and security testing.

Againn for, consenting testing there are two divisions: designer testing and deft acknowledgment testing. Them two are the being customized all through the lifecycle to encourage constant debilitating testing.

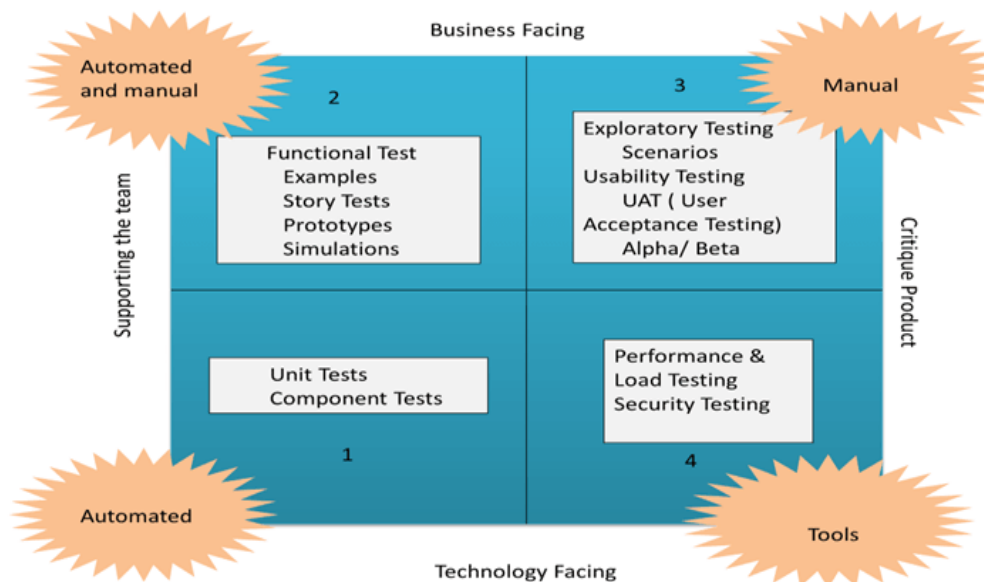
Lithe affirmation testing is a mix of ordinary utilitarian testing and standard affirmation testing as the item headway gathering, and accomplices are required to in general do it. In spite of the way that architect testing is a mix of customary unit testing and ordinary organization coordination testing, engineer testing affirms both the application code and the database arrangement.

3. Transition Phase

The objective of "Discharge, End Game" stage is to change over the framework into its creation in an effective way. The exercises which are incorporated into this fragment are working out of end clients, bolster individuals and operational individuals. Likewise, it incorporates promoting of the item discharge, back-up and reclamation, conclusion of framework and client documentation.

The last testing stage incorporates total framework testing and acknowledgment testing. So as to end a definitive phase of testing with no deterrents, one ought to need to explore the item more completely while it is in erection cycles. Amid the closure diversion, analyzers will chip away at the imperfection stories.

4. Construction



Agile Quadrant I

In this quadrant, the inner code quality is the main spotlight and it consists of test cases which are technology driven and are implemented to bring the team to the best working.

It includes:

1. Unit .Testing
2. Componentt Testing

Agile Quadrant II

It contains test cases which are business driven and are implemented to support the agile team. This Quadrant focuses on the requirements.

It includes:

1. Testing of examples of possible scenarios and workflows
2. Testing of User experience such as prototypes
3. Pair testing

Agile Quadrant III

This quadrant give input to quadrants I and II. The experiments can be utilized to perform robotization testing. In this quadrant, numerous rounds of cycle surveys are completed which manufactures trust in the item.

It incorporates:

1. Ease of use Testing
2. Exploratory Testing
3. Pair testing with clients
4. Collective testing
5. Client acknowledgment testing

Agile Quadrant IV

This quadrant centers around the non-practical necessities, for example, execution, security, steadiness, and so on. With the assistance of this quadrant, the application is made to convey the non-utilitarian characteristics and anticipated esteem.

It incorporates:

1. Non-utilitarian tests, for example, stress and execution testing

2. Security testing w.r.t confirmation and hacking

3. Framework testing

4. Information movement testing

5. Adaptability testing

6. Burden testing

Difficulties with spy programming advancement

1. Odds of blunder are more as documentation is given less need which at last puts more weight on the QA group.
2. New highlights are presented rapidly, which decreases the accessible time for test groups to recognize whether the most recent highlights are as indicated by the necessity of the partners or clients and does it genuinely address the tailored suits.
3. Analyzers are frequently required to assume a semi-engineer job.
4. Test execution cycles are profoundly packed. Less time to get ready test plan.
5. For regression testing, they will have minimal timing.
6. Change in their role from being a gate-keeper of quality to being a partner in Quality.
7. Requirement changes and updates are inherent in an agile method, which is proving to be the biggest challenge for QA.

Danger of Automation in Agile Process

1. Computerized UI gives an abnormal state of certainty, however they are moderate to execute, delicate to keep up and costly to manufacture. Robotization may not essentially improve test efficiency except if the analyzers realize how to test a specific application.

2. Inconsistent tests are a noteworthy worry in computerized testing.

3. On the off chance that the computerized tests are started physically instead of through Continuous Integration, at that point there is a hazard that they probably won't run and subsequently may cause falling flat of tests.

4. A large portion of the accessible computerization apparatuses give straightforward highlights like robotizing the catch and replay of manual experiments. Such apparatus empowers testing through the UI and prompts a naturally hard and hard to look after tests.

5. Ordinarily mechanization test plan is half-baked so as to spare time which results in the disappointment of experiments.

6. Efficiency measurements which incorporate number of experiments made or executed every day can be deluding, and could prompt making an expansive interest in running pointless experiments.

7. Individuals from the nimble computerization group must be congenial, helpful, and clever, generally the framework may come up short.

8. Robotization may propose and convey testing arrangements that require an excessive amount of progressing support in respect to the esteem gave.

9. Robotized testing may miss the mark on the aptitude to consider and pass on effective courses of action.

10. Robotized testing may be effective to the point that they miss the mark on huge issues to disentangle, and thusly swing to unimportant issues.

Fundamentally, Communication between the gatherings makes adroit testing achievement!!!

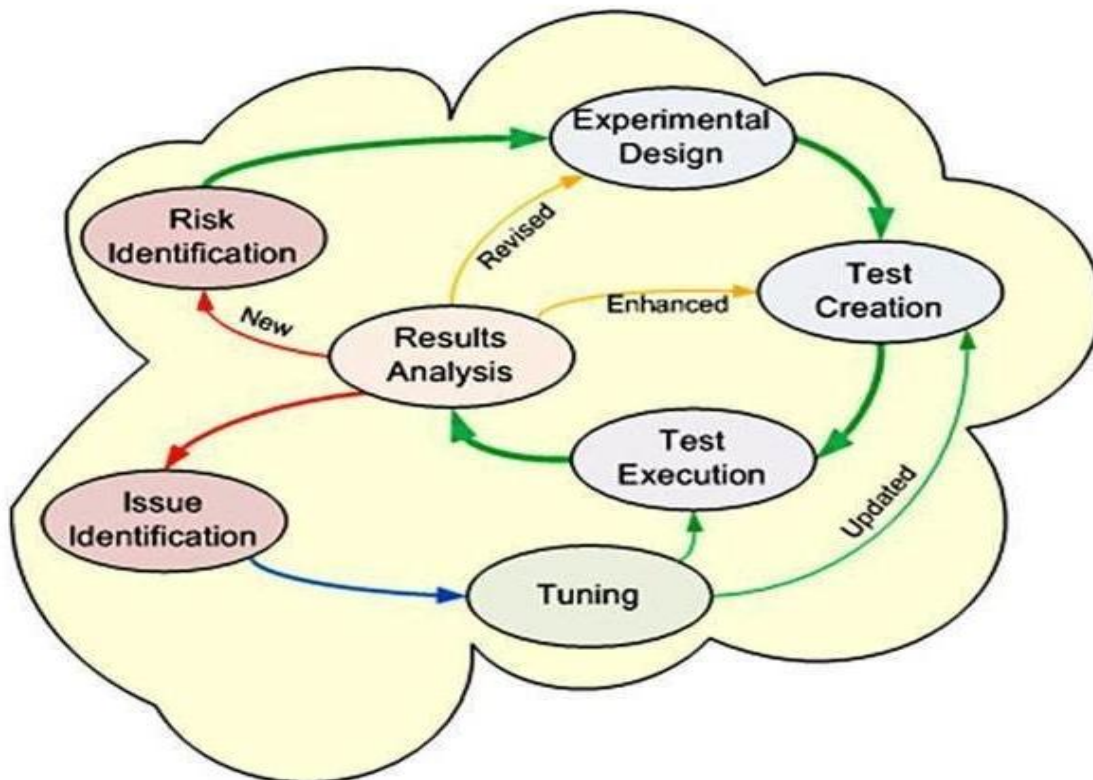
Chapter-6 RESULT AND PERFORMANCE ANALYSIS

Performance and Result for Agile Development

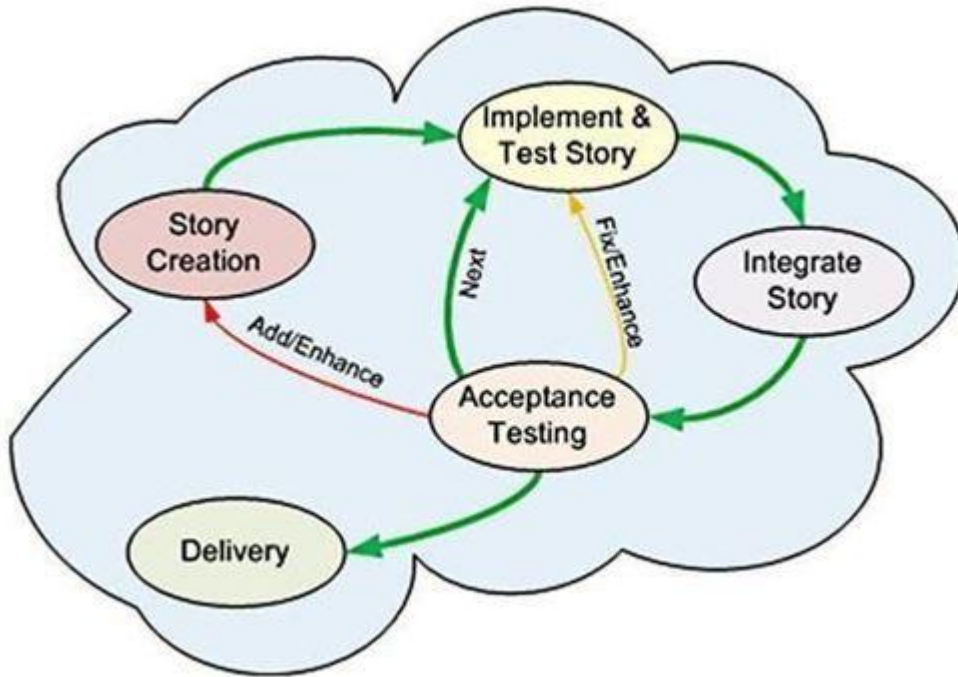
Performance testing is considered to be an integral part of agile model development. It mainly helps in development of higher quality software in less period of time while reducing down the development costs. The aim is to enhance the performance quality and reduce down the development effort and to test functionality and performance in the same sprint. More time taken to conduct performance testing results in more expense.

Performance testing cycle

Performance testing is inherently Agile. This is mainly due to its continuing, iterative feedback loop/cycle. Every performance test results in one of three states: no new information, new issue raised, or new risk identified with the next performance activity.

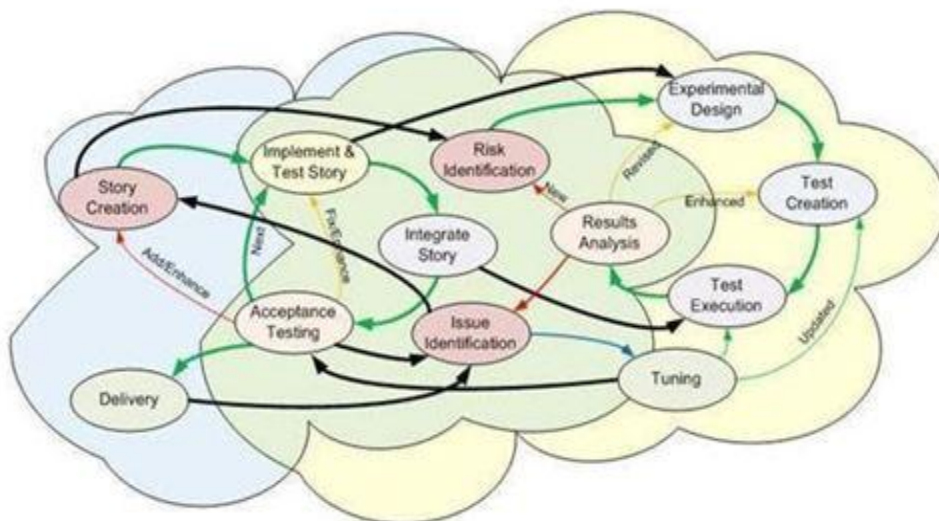


Depiction of a generic Agile software development model



Visual depiction of a generic Agile Development cycle

Integrating performance testing into an existing Agile development model



Three approaches witnessed during project work:

On demand, On retainer, and Full immersion.

On demand

Making use of “on demand” services periodically to keep project updated as per user requirement.

- Performance targets, goals, budgets must become a standard part of discussion.
- Developers must be made responsible for testing unit-level, component-level, integration-level.
- A single-user performance testing and tuning can be carried out.
- A member of the team must take given an overall responsibility for managing all the performance related tasks.

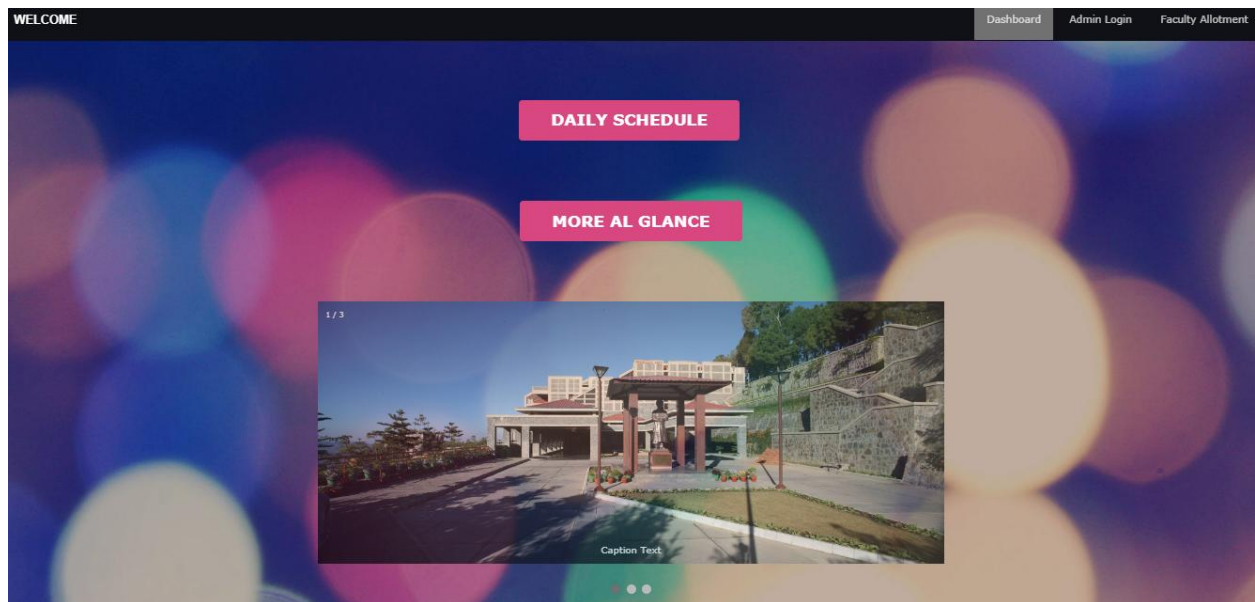
On retainer

Here each development project is assigned a specific performance tester. In this model the performance testing expertise brings in more project-level knowledge in the field of performance testing.

Full immersion

The goal for the project completion in an Agile enterprise, is mainly its performance. It is important for the value of the software. In the “full immersion” model, there are full-time member of the team who has a particular specialty in both delivering and testing performance. They are primarily responsible for coordinating and managing performance related activities throughout the entire development lifecycle(at times even throughout the entire product lifecycle).

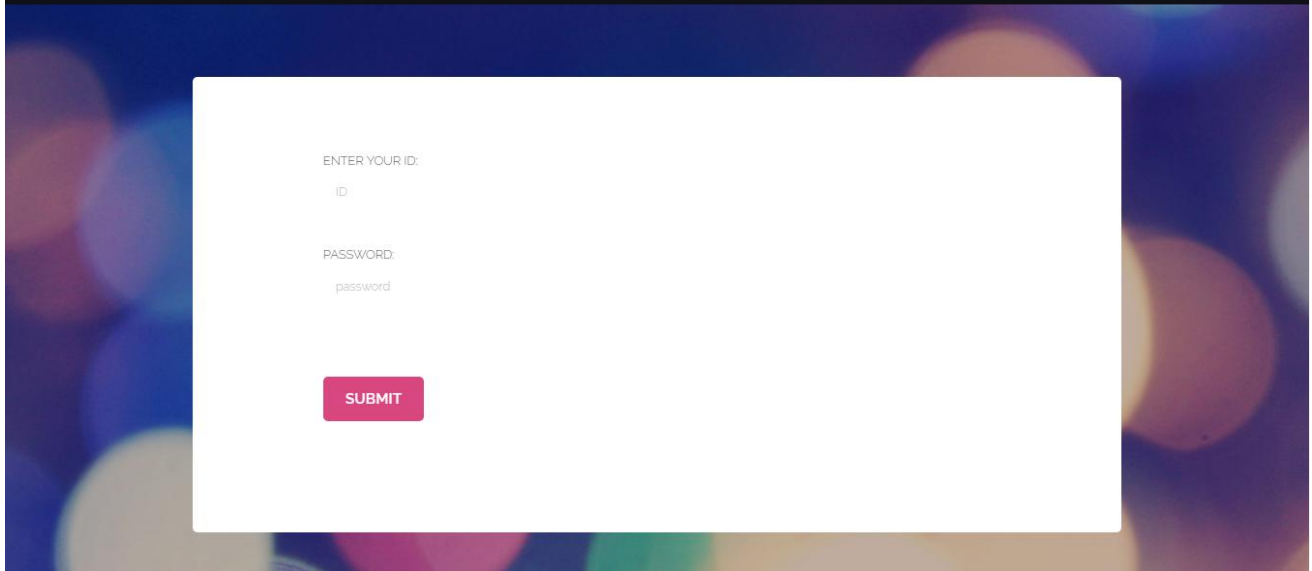
Output Depicting Screenshots:



Cover Page

*****B.Tech 2 Sem*****									
Day	9-10	10-11	11-12	12-1	1-2	2-3	3-4	4-5	5-6
Mon	L-18B11MA211 CS21-CS23 (NKT) CR1	L-18B11PH211 CS21-CS23 (RAJ) CR1	L-18B17CI211 CS21-CS23 (RBT) CR1	L-18B11EC211 CS24-CS26 (MSD) CR2	L-18B11PH211 CS27-CS28,IT21 (DSA) LT1	L-18B17CI211 CS27-CS28,IT21 (EGA) LT1			T-10B11EC211 BACK (PRG) CR6
	L-18B11MA211 EC21,CE21-CE22 (PKP) CR3	L-18B11MA211 CS24-CS26 (KAS) CR2	L-18B11PH211 CS24-CS26 (SKK) CR2	L-18B11CI211 BT21,BI21 (KSA) CR4		L-18B17CI211 CS24-CS26 (PKG) CR2	P-18B17EC271 CS25 (MSD,CHR) ECL2	P-18B17EC271 CS25 (MSD,CHR) ECL2	
	L-10B11CI211 BACK (RIV) CABIN1	L-18B11PH211 EC21,CE21-CE22 (SKT) CR3	L-18B11MA212 BT21,BI21 (MDS) CR4	T-18B11EC211 CS21 (VKB) TR3		L-18B11PH212 BT21,BI21 (RRS) CR9	P-18B17CI271 CS23 (RSU) CL3_1	P-18B17CI271 CS23 (RSU) CL3_1	
		L-18B11EC212 BT21,BI21 (HSL) CR4	T-18B11EC211 IT21 (NTJ) TR3			T-18B17CI211 CS23 (RBT) TR3	P-18B17CI271 IT21 (RIV) CL3_2	P-18B17CI271 IT21 (RIV) CL3_2	
	P-18B17CI271 CS28 (SUM) CL5_1	P-18B17CI271 CS28 (SUM) CL5_1	T-18B11MA211 CS28 (PKP) TR4			T-18B11MA211 CS22 (NKT) TR4	P-18B17CI271 BI21 (VSG) CL5_1	P-18B17CI271 BI21 (VSG) CL5_1	
	P-18B17PH271 IT21 (DSA) PHLAB1	P-18B17PH271 IT21 (DSA) PHLAB1	T-18B17CI211 CE22 (RSU) TR5			T-18B11EC211 EC21 (PRG) TR5	P-18B17PH271 CS24 (SKT) PHLAB1	P-18B17PH271 CS24 (SKT) PHLAB1	
	P-18B17GE172 CS27 (TNM) DRAWR	P-18B17GE172 CS27 (TNM) DRAWR	P-18B17GE172 CS27 (TNM) DRAWR			P-18B17GE171 CE22 (AGA) WORKLAB	P-18B17GE171 CE22 (AGA) WORKLAB	P-18B17GE171 CE22 (AGA) WORKLAB	
			P-18B17EC271 CE21 (PRG,SKH) ECL1	P-18B17EC271 CE21 (PRG,SKH) ECL1		P-18B17GE172 CS21 (ADP) DRAWR	P-18B17GE172 CS21 (ADP) DRAWR	P-18B17GE172 CS21 (ADP) DRAWR	
			P-18B17CI271 EC21 (EGA) CL3_1	P-18B17CI271 EC21 (EGA) CL3_1			T-18B17CI211 CS27 (EGA) TR3		
							T-18B11EC211 CS22 (VKB) TR4		
Tue	L-18B17CI211 CS27-CS28,IT21 (EGA) LT1	L-18B11MA211 CS27-CS28,IT21 (PKP) LT1	L-18B11EC211 CS27-CS28,IT21 (NTJ) LT1	T-10B11EC211 BACK (PRG) CR6	L-18B17CI211 EC21,CE21-CE22 (RSU) CR3	L-18B11EC211 EC21,CE21-CE22 (PRG) CR3			
	L-18B11EC211 CS21-CS23 (VKB) CR1	L-18B11MA211 CS21-CS23 (NKT) CR1	T-18B11EC211 CS25 (MSD) TR3		T-18B11PH211 BACK (DSA) TR5	T-18B17CI211 CS21 (RBT) TR3			
	T-18B11MA211		T-18B11EC211 CE21		T-10B11PD211 BACK (PRG)	T-18B11MA211			

Daily Scheduled time table showing B.tech and M.Tech (all semesters) lectures

The image shows a web page for an admin login. It features a dark blue header with navigation links. The main content area has a colorful bokeh background. A white login form is centered, containing two input fields for ID and password, and a red submit button.

ENTER YOUR ID:

ID

PASSWORD:

password

SUBMIT

Admin Login Page

FORM

KINDLY FILL THE FORM

Enter semester:

SEMESTER

Enter Course Code :

COURSE CODE

Enter your name:

NAME

Enter your Employee id:

EMPID

Subject:

SUBJECT

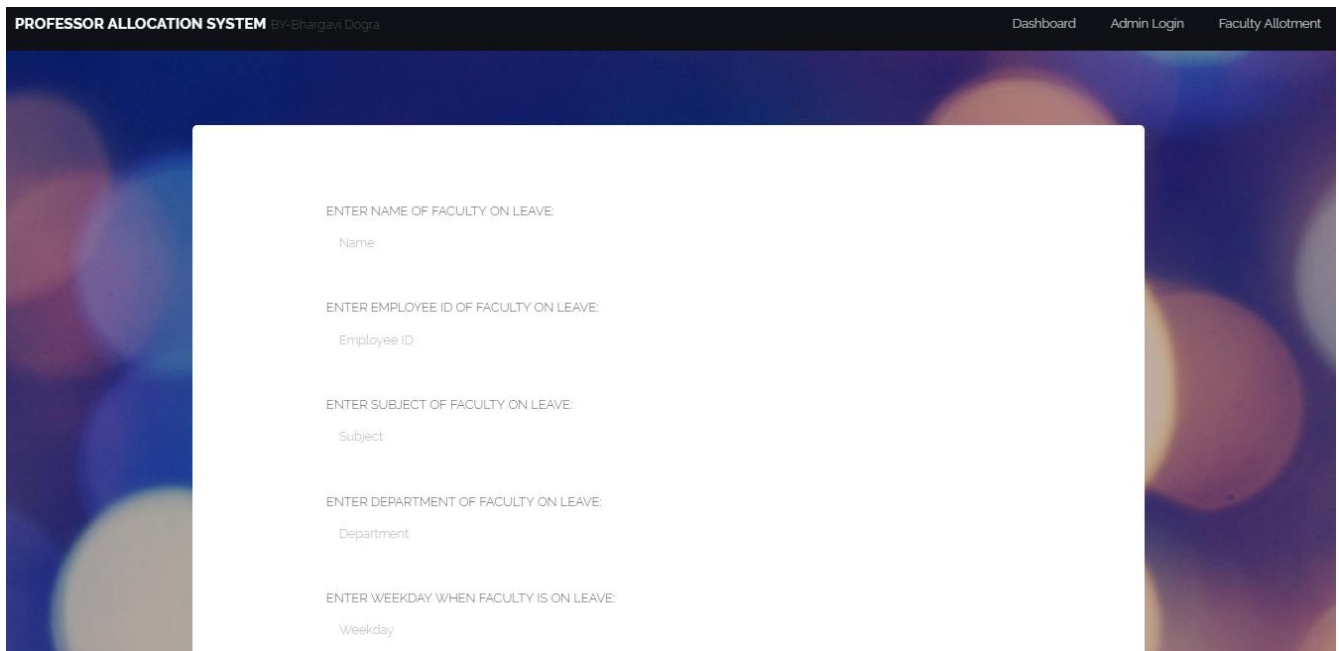
Department:

Department

Enter week day:

week day

Form to enter professor's lecture schedule



Absentee Faculty's lecture allotment portal



Example of lecture allotment to other faculty member

row(s) starting from record # 390
 cells
 Page number: 13

+ Options		Sem	Course Code	name	empid	subject	department	day	nine	ten	eleven	twelve	one	two	three	four
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1611	Sh Prateek	PTK	Computer Graphics	CSE	TUESDAY	1	0	1	0	0	0	1	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1611	Sh Prateek	PTK	Computer Graphics	CSE	WEDNESDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1611	Sh Prateek	PTK	Computer Graphics	CSE	THURSDAY	1	0	0	0	0	0	1	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1611	Sh Prateek	PTK	Computer Graphics	CSE	FRIDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1611	Sh Prateek	PTK	Computer Graphics	CSE	SATURDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1671	Sh Himanshu Jindal	HJL	Computer Graphics Lab	CSE	MONDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1671	Sh Himanshu Jindal	HJL	Computer Graphics Lab	CSE	TUESDAY	0	0	0	0	0	0	1	1
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1671	Sh Himanshu Jindal	HJL	Computer Graphics Lab	CSE	WEDNESDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1671	Sh Himanshu Jindal	HJL	Computer Graphics Lab	CSE	THURSDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1671	Sh Himanshu Jindal	HJL	Computer Graphics Lab	CSE	FRIDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1671	Sh Himanshu Jindal	HJL	Computer Graphics Lab	CSE	SATURDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1611	Sh Himanshu Jindal	HJL	Computer Graphics	CSE	MONDAY	1	0	0	0	0	0	0	0
<input checked="" type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1611	Sh Himanshu Jindal	HJL	Computer Graphics	CSE	TUESDAY	0	0	1	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1611	Sh Himanshu Jindal	HJL	Computer Graphics	CSE	WEDNESDAY	1	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1611	Sh Himanshu Jindal	HJL	Computer Graphics	CSE	THURSDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1611	Sh Himanshu Jindal	HJL	Computer Graphics	CSE	FRIDAY	1	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	11B1WC1611	Sh Himanshu Jindal	HJL	Computer Graphics	CSE	SATURDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	10B17C1672	Dr Rajni Mohana	RJM	Compiler Design Lab	CSE	MONDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	10B17C1672	Dr Rajni Mohana	RJM	Compiler Design Lab	CSE	TUESDAY	0	0	1	1	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	10B17C1672	Dr Rajni Mohana	RJM	Compiler Design Lab	CSE	WEDNESDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	10B17C1672	Dr Rajni Mohana	RJM	Compiler Design Lab	CSE	THURSDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	10B17C1672	Dr Rajni Mohana	RJM	Compiler Design Lab	CSE	FRIDAY	0	0	1	1	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	10B17C1672	Dr Rajni Mohana	RJM	Compiler Design Lab	CSE	SATURDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	10B11C1612	Dr Rajni Mohana	RJM	Compiler Design	CSE	MONDAY	0	0	0	1	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	10B11C1612	Dr Rajni Mohana	RJM	Compiler Design	CSE	TUESDAY	1	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	10B11C1612	Dr Rajni Mohana	RJM	Compiler Design	CSE	WEDNESDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	10B11C1612	Dr Rajni Mohana	RJM	Compiler Design	CSE	THURSDAY	1	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	10B11C1612	Dr Rajni Mohana	RJM	Compiler Design	CSE	FRIDAY	0	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	10B11C1612	Dr Rajni Mohana	RJM	Compiler Design	CSE	SATURDAY	1	0	0	0	0	0	0	0
<input type="checkbox"/>	<input type="checkbox"/>	B6	10B17C1672	Dr Pradeep Kumar Singh	PKS	Compiler Design Lab	CSE	MONDAY	0	0	0	0	0	0	0	0

/ With selected:

Backend database Glimpse

Faculty Allotment of lectures for Sh Himanshu Jindal

Faculty does not have lecture scheduled between (9-10) :

Faculty does not have lecture scheduled between (10-11) :

Names of faculty members available for lecture between (11-12) :
Sh Nitin Kumar

Faculty does not have lecture scheduled between (12-1) :

Faculty does not have lecture scheduled between (1-2) :

Faculty does not have lecture scheduled between (2-3) :

Faculty does not have lecture scheduled between (3-4) :

Faculty does not have lecture scheduled between (4-5) :

End result depicting the lecture being allotted to other faculty member

Chapter-7 CONCLUSION

7.1 Outcome

It has been previewed with a fact that the application deployed works according to the requirements and expectations. It possesses user friendliness attribute and is cost efficient as well. Testing has been performed on our software thoroughly and application is debugged properly to eliminate all sorts of errors arising. The application is completely able to integrate all the user entered utility services all at one place. The deployed app makes use of Model, View and Controller architecture in order to enhance its output utility.

7.1 Future Scope

Each and every project always has a scope for more improvements. The demonstrated project has provision yet to be enhanced and improved via following ways:

The UI is completely enhanced to make it more friendly for the users and to increase the responsiveness.

The unlocking functionality of the account can be embedded in the software for extending towards password change as an option.

Email facility for providing important notifications to the employees.

Report any attempted threats to security mechanism if arising.

Employee /Faculty dashboard and Resolution functionality can be added as well.

